Legal Knowledge and Information Systems
S. Villata et al. (Eds.)

doi:10.3233/FAIA200847

Events Matter: Extraction of Events from Court Decisions

Erwin FILTZ [a,b], María NAVAS-LORO [c], Cristiana SANTOS [d], Axel POLLERES [a] and Sabrina KIRRANE [a]

[a] *Vienna University of Economics and Business*
[b] *Siemens AG Österreich*
[c] *Universidad Politécnica de Madrid – Ontology Engineering Group, Madrid, Spain*
[d] *Utrecht University*

Abstract. The analysis of court decisions and associated events is part of the daily life of many legal practitioners. Unfortunately, since court decision texts can often be long and complex, bringing all events relating to a case in order, to understand their connections and durations is a time-consuming task. Automated court decision timeline generation could provide a visual overview of what happened throughout a case by representing the main legal events, together with relevant temporal information. Tools and technologies to extract events from court decisions however are still underdeveloped. To this end, in the current paper we compare the effectiveness of three different extraction mechanisms, namely deep learning, conditional random fields, and rule-based method, to facilitate automated extraction of events and their components (i.e., the event type, who was involved, and when it happened). In addition, we provide a corpus of manually annotated decisions of the European Court of Human Rights, which shall serve as a gold standard not only for our own evaluation, but also for the research community for comparison and further experiments.

Keywords. event extraction, named entity recognition, court decisions

1. Introduction

Court decisions are an important source of law information for legal practitioners: they elaborate on the facts of a case, involved parties, interpretations of the circumstances, the applicable law and legal principles, and finally the legal assessment leading to the decision. Legal professionals constantly extract, interpret and reason with and about prior cases whilst arguing for a decision in a current, undecided case. However, court decisions texts can be long and complex and thus time-consuming to read. Therefore it would be beneficial to find a means to provide a quick overview of a case, thereby helping to turn decisions into operational, consumable and actionable legal knowledge.

In this work we focus specifically on using Natural Language Processing (NLP) techniques to automatically extract the essence of a court case. Besides extracting general legal rules from individual cases, we aim at providing a quick overview of what happened, who was involved and when the event took place. In the terminology of NLP, event extraction can be treated as a *text classification task* aiming at assigning text fragments (typically, paragraphs, sentences or smaller parts of documents) to predefined (event)

classes [1]. Another, related NLP task is Named Entity Recognition (NER) which extracts entities referred to in texts and classifies them into categories [2], for instance people, places and organizations; moreover, named entities can also be domain-specific, for instance, courts or laws. Event extraction can benefit from NER, since it can be used to enrich events with relevant information, such as the parties involved. This paper focuses on the extraction of events and their components from court decisions of the European Court of Human Rights (ECHR)[1] based on a sample thereof.

Summarizing our *contributions*, we: (i) provide a corpus of manually annotated ECHR decisions; (ii) perform a comparison of different approaches to automatically extract events and their components – implementations as well as our evaluation results are made available on GitHub; and (iii) introduce a prototypical web interface that can be used to display court decisions along with their extracted timelines.

The remainder of this paper is structured as follows. We outline related works in Section 2. Our corpus as well as the annotation methodology is described in Section 3. Section 4 contains information about the compared event classification and NER approaches, followed by Section 5 discussing evaluation results. Section 6 provides conclusions.

2. Related Work

Recent advances in NLP are often based on embedding text in multidimensional vector space, with neural network architectures being trained on such numeric representations. Such methods yield in re-usable, publicly available language models trained on large corpora of texts, where embeddings can be created on different levels, for instance words, sentences and documents. While pre-training models on large corpora of generic texts is a very time-consuming process [3], adapting (aka fine-tuning) such generic models to domain-specific language is often less demanding.

Overviews on diverse automated event extraction approaches in the general domain can be found in literature [4,5]. Specifically in the legal domain [6], existing work usually involves searching for *ad hoc* definitions of events, ignoring general event annotation schemas such as the ACE 2005 model [7]. Several approaches tend to be supported by patterns, using manually crafted rules or semantic role labeling techniques [8,9,10,11]. Other approaches do not search for events specifically, but target legal case factors [12].

The automated generation of timelines out of annotated documents could help to get a better and faster understanding of the content of a document. Existing work focusing on this task include Linea [13], a system that is able to build and navigate timelines from unstructured text, and TimeLineCurator [14] a system that is primarily designed to allow journalists to generate temporal stories, however can be used to produce a timeline from any free text or url. Furthermore, the creation of timelines has also been investigated in other domains, such as medicine [15,16] and journalism [17]. We refer to [14] for further details on the respective approaches.

Regarding corpora in the legal domain, court decisions of the ECHR have also been used in literature for different tasks [18,19]. Nevertheless, very few annotated corpora from the legal domain have been made available, and to the best of our knowledge none of them considers events.

[1] `https://echr.coe.int/`

3. Corpus and annotation methodology

This section describes the ECHR corpus as well as our annotation methodology.

Description of the corpus. The corpus consists of 30 decisions of the ECHR. The ECHR decisions were chosen because they contain: i) different types of time-related events concerning different actors in comparison with the decisions of the Court of Justice of the EU [6]; and ii) a standard structure in which different legal events are embedded. ECHR decisions are divided into several sections containing specific information according to Rule 74 of the Rules of the Court [20]: the *Preamble* and the *Introduction* are followed by *Facts* which contain information about the formal procedure and the circumstances of the case providing details about what happened. The following *Law* section describes the legal situations and states the alleged violation(s). The document concludes with the *Decision* section. For the purposes of this paper, we use the mentioned document structure excluding the *Law* section and focus on the procedure, circumstances and decision.

Annotation methodology. The corpus was annotated by two legal experts in several iterations. The experts annotated independently and then met with a third person to reach a consensus on the disagreements. In this work, as we focus on event extraction aimed to automated court decision timeline generation, we were interested in information that is relevant to searching for or extracting time-related information, such as events, processes, temporal information, and the parties involved. As time-related events of cases are linguistically expressed, we annotated the most salient candidate passages thereof. The decisions were manually annotated following the frame "who-when-what". To illustrate the applicability thereof, we make use of an annotated paragraph of the case Altay v. Turkey (no. 2), no. 11236/09, 9 April 2019 (a case referring to respect of private life):

"On 29 May 2008 the applicant lodged an application with the Edirne Enforcement Court for the restriction on the conversations between him and his lawyer to be lifted."

"Who" corresponds to the subject of the event, which can either be a subject, but also an object (i.e., an application); in the example, the subject is "(the) applicant". *"When"* refers to the date of the event, or to any temporal reference thereto; in the paragraph considered, the "when" is the "29 May 2008". *"What"* usually corresponds to the main verb reflecting the baseline of all the paragraph (which in this case is "lodged"); additionally, we include thereto a *complementing* verb or object whenever the core verb is not self-explicit or requires an extension to attain a sufficient meaning; in the paragraph considered, the "what" is "lodged an application". Another e.g. is "dismiss an action". *"Event"* relates to the extent of text containing contextual event-related information. The *type* of such annotations can be either *circumstance* – meaning that the event correspond to the facts under judgment; or *procedure*– wherein the event belongs to the procedural dimension of the case. This includes court procedures (legal proceedings stricto sensu), but also actions that trigger procedural effects. A further analysis of this distinction can be found in previous literature [6,18]. In the paragraph at stake, we annotated as *event* the whole sentence, being its type *procedure*. Further, we have annotated events and their temporal dimension (related-time events) with concrete guidelines:

Extension of what event element. One *what* event element can also include two or more close-related verbs, e.g. "divorced" and "agree on custody", instead of annotating two connected verbs autonomously. Moreover, whenever there is some causal relationship between events, we annotate merely one, e.g. “they drink water and they felt unwell".

Repeated events. When there is reference to events happening in several dates (e.g. “the dates of birthday of three applicants, respectively“), we annotate solely one event as the *what*, and add just one annotation that covers all the related dates.

Non-dated events. Events that are not dated, though semantically expressing an implicit time reference, are then annotated under “when”, for example, the time expressions as “the same date”, “this afternoon”, “on unspecified dates”, “in a number of occasions”.

Non-annotated events. Some events were not considered relevant to be depicted in a timeline, and therefore not annotated, e.g. the fact that *X was born in X* seemed irrelevant.

Factuality. Events that are mentioned in the text but do not occur, are yet annotated with the feature “factuality”, but not included in the timeline. When events are negated, this feature equals to “NOT”, for instance, a party does not appeal against a decision.

Difficult and blurred annotations. During the annotation process, some events were difficult to tag, and others sparked discussion about how to do it, challenging the stipulated guidelines and evidencing how complex and subjective annotating tasks can be. Due to space constrains, we only show one sample annotation that triggered discussion on the type of events between procedure/circumstance. Further examples can be found in the corpus webpage. Regarding the paragraph *"On 26 February 2014 the Deputy Town Prosecutor carried out an inspection of remand prison SIZO-6"*, the issue relates to the semantics attributed to the role "Deputy Town Prosecutor" which renders the idea of being a court magistrate, and as such, it would be deemed as a procedural event. Herein, the function instead refers to an inspection task, without procedural effect.

4. Event extraction and named entity recognition

Herein we describe different methods used in our experiments for the extraction of events and their components in the ECHR court decisions. The applied approaches include deep-learning- and embeddings- based, conditional random fields and rule-based methods. The corpus and the code used in this paper is available on Github[2].

4.1. Deep learning

The task of assigning one or multiple classes from a set of classes to a text fragment is called text classification [1]. Fragments in our context are typically sentences that are classified into the types *procedure*, *circumstance* or neither. Hence we deal with a multi-class classification problem. The extraction of the event components is similar to a Named Entity Recognition Problem. We use a state-of-the-art model as a baseline and compare it further with additional approaches selected upon their results on legal texts (cf. [21,22, 23]). As there is no pre-trained legal model available, we apply the common approach

[2] `https://mnavasloro.github.io/EventsMatter/`

of *fine-tuning* a Universal Language Model for Text Classification (ULMFiT) [3] which takes a generic model and tunes it with a domain-specific corpus (called transfer learning). In terms of preprocesing, we remove very short sentences from the dataset, for instance headings such as *II THE LAW*. The models are:

Flair and Flair-finetuned. We first selected the generic *news-forward-fast* language model from the Flair embedding approach [24], which is pre-trained on a corpus with one billion words as our baseline model. We also fine-tune the pre-trained model with the documents from our corpus for one epoch (which took more than seven hours).

Flair ECHR. There are no specific legal pre-trained models available that we could use for our experiments. On a different classification task, we made good experiences in prior work with using a domain specific model trained on a small corpus of EU legal documents outperforming generic models in a multi-label text classification task [25]. Therefore, we also train a model on a corpus of 13,000 ECHR court decisions acquired from the European Court of Human Rights OpenData project [26] for four epochs.

BERT and BERT-finetuned. The Bidirectional Encoder Representations from Transformers (BERT) [27] is a language model learning the context of words in a bidirectional way and is applicable to many tasks. We use a BERT model (*bert-base-cased*) pre-trained on Wikipedia and a book corpus, plus further add a layer on top fine-tuning the model with the ECHR corpus for two epochs.

DistilBERT and DistilBERT-finetuned. DistilBERT [28] is a lightweight version of BERT that makes use of a teacher-student setup to distill the knowledge of the large model (BERT) to the student (DistilBERT). Our fine-tuned model (two epochs) is based on the pre-trained *distilbert-base-cased* model with an additional ECHR corpus layer.

4.2. Conditional Random Fields

Conditional Random Fields (CRF) are used for the mapping of sequences based on probabilistic models to label sequences [29]. CRF have already been applied in similar tasks in the legal domain for extracting specific legal entities, such as lawyers, courts and legal literature [30]. A CRF model uses features of a token, for instance casing, position of the token and subsequences, to calculate the probability that it is preceded or followed by a particular other token. It also takes the probabilities into account that a specific named entity, for instance a temporal information is followed by a subject.

4.3. Rule-based

Unlike the previous approaches, implemented as a classification task, the rule-based approach is an annotation task based on a search for specific patterns of events in the form of frames. Our approach has two steps: i. the collection of frames (done before the annotation), and ii. the event extraction that uses the frames in order to annotate a text.

1. Frame collection. We listed all *what* event components in the training set, and then identified the main verb, its type and the dependency relations (using the CoreNLP dependency parser [31]), within the *what*, and towards the subject (*who*), including the object for both possible active and passive voices since they are very different. When there are several mentions of the same main verb, all information is gathered and combined

into a single frame. Once all the *what* elements are processed, they are stored for later use by the extraction algorithm.

2. Event extraction. Using the previously obtained frames, we look for the relevant events in the text. Since there are events that can appear many times in a text, we just consider events that have a date attached. To find dates and their normalized value (in order to be able to build a timeline), we adapted the Añotador software [32]. Then we used the information from the frames to look for the main verb of the event and for the previously identified dependency relations, as well as some Part-of-Speech considerations (using also CoreNLP). Additionally, some specific events that tend to appear always in the same form in the text (such as the final decision) are identified using regular expressions.

4.4. Use case: Timeline generation

In order to enable an intuitive way to overview a case, we decided to generate timelines from the case. We developed a demonstrator [3] that takes the *id* of a ECHR case and returns its rule-based annotation and generates a timeline. Through this timeline, we can navigate a case going directly to the event mention in the text just by clicking on it in the timeline. The fact that it directly refers to the text allows the user to retrieve the context of the event, as well as surrounding information that might not be reflected in the timeline.

5. Evaluation and Discussion

In this section we present results of our experiments. For experiments based on deep learning approaches, we used the state-of-the-art NLP library Flair[4] which uses contextualized string embeddings (called FlairEmbeddings) that captures the semantics and the context, and therefore, produce different context dependent embeddings for the same words [24]. The pre-trained transformer models (BERT, DistilBERT) are provided by the Huggingface library [33] and can be easily imported into Flair. The Flair ECHR model is created using the Flair library, and fine-tuning of the BERT and DistilBERT models is also based on the transformers library by Huggingface. All models have been trained with the same settings of a maximum of 150 epochs, patience of 3 and an anneal factor set to 0.5 and the training is automatically stopped when the learning rate is too small. We use common evaluation metrics: *Precision (P)*, *Recall (R)* and *F-score (F)*.

The documents have an average size of 2,302 tokens without the legal section (legal framework). Each document includes on average 21 different events, divided into 10 *procedure* and 11 *circumstance* events on average. The number of *who* occurrences amounts to 13.9 on average, while the number of temporal information annotations (*when*) to 17.6, and the number of *core* annotations to 24. We split the dataset into training, testing and validation set on a document level applying 5-fold cross-validation (in the deep learning based approach) such that the training set consists of 24, and the test and validation set of three documents each. The results represent the average of all splits. The results for all approaches are presented in Table 1. When comparing different approaches on event (component) extraction, we can observe that more advanced language models based on

[3] `https://whenthefact.oeg-upm.net/`
[4] `https://github.com/flairNLP/flair`

Table 1. Evaluation results for event classification and event components. (*P=Precision, R=Recall, F=F-score. Best results highlighted in boldface.*)

		Event Types		Event Components		
		Procedure	Circumstance	What	When	Who
CRF	P	82.39	68.78	85.10	89.30	89.09
	R	80.26	47.88	76.91	84.46	70.38
	F	80.80	54.78	**80.50**	86.58	78.34
Flair pretrained	P	83.32	57.21	56.41	90.50	89.93
	R	78.95	32.64	45.50	79.65	76.49
	F	80.31	40.57	50.10	84.35	82.30
Flair finetuned	P	87.07	58.88	60.12	90.87	91.63
	R	81.57	51.12	51.79	80.02	83.71
	F	84.13	53.33	55.58	84.87	87.44
Flair ECHR	P	76.78	41.93	57.94	82.00	40.48
	R	71.21	13.12	15.69	57.88	11.87
	F	73.86	17.92	23.28	66.88	18.23
BERT pretrained	P	81.95	66.70	60.45	85.88	86.37
	R	80.79	49.23	61.17	88.22	89.90
	F	80.56	54.31	60.78	86.98	88.05
BERT finetuned	P	91.44	76.81	65.58	89.45	88.88
	R	90.20	78.94	66.26	91.01	92.22
	F	90.55	77.59	65.83	**90.22**	**90.44**
DistilBERT pretrained	P	83.91	56.53	59.58	81.87	86.67
	R	83.57	51.63	57.45	86.35	85.73
	F	83.26	53.26	58.41	83.95	86.09
DistilBERT finetuned	P	91.64	81.61	62.79	87.31	89.92
	R	93.27	78.65	62.06	89.33	90.12
	F	**92.38**	**79.75**	62.37	88.23	89.98

		Event				Event Components					
		Identification		Type		What		When		Who	
		Len	Str	Len	Str	Len	Str	Len	Str	Len	Str
Rules	P	85.71	80.00	47.14	42.86	80.26	23.68	77.59	72.41	75.00	68.75
	R	77.92	72.73	42.86	38.96	69.32	20.45	63.38	59.15	63.16	57.89
	F	81.63	76.19	44.90	40.82	74.39	21.95	69.77	65.12	68.57	62.86

the transformer architecture [34] (BERT and DistilBERT), in general, provide a better result compared to the standard embedding models (Flair). Furthermore, we can see that the application of the ULMFiT approach to finetune generic language models, with a domain-specific corpus, leads to improved results between less than 1% (Flair pretrained to Flair finetuned for *who*) and 25% (DistilBERT for *circumstance*). The average increase in performance with fine-tuning is 8% for recognizing *procedure* and 21% for *circumstance* events, resp. The results of the CRF approach for the *what* component is unexpected, as it outperforms the more advanced methods by approximately 20%. The results for the extraction of the event components show that recognizing temporal information (*when*) of an event yields better results than the *what* of an event by 27% and the subject (*who*) by 21% (averaged over all approaches). The performance increase for the extraction of the event components of fine-tuned models, compared to generic models, is with 5% (what), 3% (when) and 4% (who) lower compared to the results for event types.

We see that results within the event type detection are within approx. 20% over all approaches, with the worst result being achieved by the Flair ECHR approach (F 73.86%), and the best result by the DistilBERT finetuned approach with an F-score of 92.38%. The results for the *circumstance* event type show a bigger spread between the worst result of the Flair ECHR approach with an F-score of only 17.92%, while the best result is achieved by DistilBERT finetuned (F 79.75%). For the *circumstance* event types we see generally lower results than for *procedure* type detection. We attribute this to the fact that the linguistic variety of the *procedure* events is narrower as they refer to a restricted set of ways of how to express them. The performance of the Flair ECHR model showed the least performance, due to being trained only on 13,000 ECHR documents, while it is common to train language models on much larger corpora to capture the basics of a language.

The performance differences between the *procedure* and *circumstance* event classes are evident with the latter results being worse by 29% on average. *Procedure* events capture formal processes throughout a legal trail and the ways to formulate the same events is somewhat restricted, for instance, *the court upheld the judgment*; in the description of the *circumstance*s of a case, however, the English language is potentially used in its entirety. Similarly, we observe the same behavior with the results for the event components with the results for *when* and *who* being better than the results for *what*. We attribute this to the fact that absolute temporal information (e.g. a date) contained in the court decisions under investigation always follows the structure of *Day Month Year*, and the number of acting subjects is also limited to a certain range of persons (eg. applicant, judge, prosecutor), authorities (eg. Supreme Court, housing authority) or things (eg. application, appeal). Relative temporal information (eg. *X days later*, *between X and Y* or *until X*) is also expressed in a few ways only.

Overall, we can say that fine-tuning an existing language model trained on a large corpus that captures the basic features of a language with a domain-specific corpus performs better than training a new language model with a rather small domain-specific corpus. Moreover, the more restricted the variety of class candidates for classification is, the better the results. The same applies to the information following a specific format, i.e. temporal information in the form of dates.

Regarding the rule-based approach, the evaluation is slightly different. While in the deep learning approach (first table) the number of named entities reflect the results of finding the event arguments *only* in those sentences where there is an event. On the contrary, the rule-based approach (second table) finds the events and the arguments in the same algorithm, so the results of the argument are contingent upon the event results. Additionally, we provide both *strict* and the *lenient* results, meaning that either the extent of our annotation match exactly to the one by the annotators or that it only overlaps (adding or omitting some words), resp. Also, the event evaluation includes finding the extent of the event, and then, over this finding, decide its type. The annotation and evaluations for the rule-based approach were done with the software GATE [35].

From the results of the rule-based approach we see that in the event finding task we got good results, both in the strict and lenient case, meaning that most of the events are correctly found and with the correct extent. Generally speaking, we identify about 4 out of every 5 relevant events, and additionally some that were not marked as relevant (although this does not mean they are not events). Regarding event types, the results for rule-based approaches are not very promising, mainly due to the fact that the same verb

can often represent both circumstantial or procedural events, depending on surrounding information that the current rule-based implementation is not able to identify.

Results for detecting event arguments with the rule-based approach, on the other hand, are very different. While the *what* event component has very bad strict results, mainly due to the difficulty to determine the extent of the relevant modifiers of a verb, the *who* and the *when* show very good results, finding correctly most of them (e.g., 68.57% of the *who* taken into account that the limit was less than the 81.63% of the events) and almost always with the correct extent. The lenient results of the core, similar to the ones from the other arguments, demonstrates that besides the extent, the identification is correct.

6. Conclusions and Future Work

This paper presented a new corpus of legal decisions annotated with relevant events, along with a comparison of different approaches for the extraction of events and their components. Moreover, we tested state of the art methods to accomplish this annotation task automatically with promising results. To illustrate the utility of this task, we implemented an online timeline generation service which could be used by lawyers to get a quick overview of a case, thereby helping to turn decisions into operational, consumable and accessible legal knowledge.

To the best of our knowledge there is no previous comparison of event extraction techniques over legal texts in literature, and neither an available legal corpus annotated with events. In future work it would be interesting to validate the results with decisions from other courts such as the European Court of Justice or the United States Supreme Court, which are structured differently.

Acknowledgements. María Navas-Loro[5] work was partially supported by a Predoctoral grant from the I+D+i program of the Universidad Politécnica de Madrid. Sabrina Kirrane is funded by the FWF Austrian Science Fund and the Internet Foundation Austria netidee SCIENCE programme.

References

[1] Sebastiani F. Machine learning in automated text categorization. ACM Comput Surv. 2002;34(1):1–47.
[2] Grishman R, Sundheim BM. Message understanding conference-6: A brief history. In: COLING 1996 Volume 1: The 16th International Conference on Computational Linguistics; 1996. p. 466–471.
[3] Howard J, Ruder S. Fine-tuned Language Models for Text Classification. CoRR. 2018;abs/1801.06146.
[4] Hogenboom F, Frasincar F, Kaymak U, De Jong F. An overview of event extraction from text. In: DeRiVE@ ISWC. Citeseer; 2011. p. 48–57.
[5] Xiang W, Wang B. A Survey of Event Extraction From Text. IEEE Access. 2019;7:173111–173137.
[6] Navas-Loro M, Santos C. Events in the legal domain: first impressions. In: TERECOM@JURIX; 2018. p. 45–57.
[7] The ACE 2005 Evaluation Plan.;. `https://api.semanticscholar.org/CorpusID:10821576`.
[8] Kiyavitskaya N, Zeni N, Breaux TD, Antón AI, Cordy JR, Mich L, et al. Automating the extraction of rights and obligations for regulatory compliance. In: International Conference on Conceptual Modeling. Springer; 2008. p. 154–168.
[9] Maxwell KT, Oberlander J, Lavrenko V. Evaluation of semantic events for legal case retrieval. In: Proceedings of the WSDM'09 Workshop on Exploiting Semantic Annotations in Information Retrieval. ACM; 2009. p. 39–41.

[5]ORCID 0000-0003-1011-5023

[10] Lagos N, Segond F, Castellani S, O'Neill J. Event extraction for legal case building and reasoning. In: International Conference on Intelligent Information Processing. Springer; 2010. p. 92–101.
[11] Navas-Loro M, Satoh K, Rodríguez-Doncel V. Contractframes: Bridging the gap between natural language and logics in contract law. In: JSAI International Symposium on Artificial Intelligence. Springer; 2018. p. 101–114.
[12] Wyner AZ, Peters W. Lexical Semantics and Expert Legal Knowledge towards the Identification of Legal Case Factors. In: JURIX. vol. 10; 2010. p. 127–136.
[13] Etiene T, et al. Linea: Building Timelines from Unstructured Text. In: 28th SIBGRAPI Conference on Graphics, Patterns and Images, SIBGRAPI 2015. IEEE Computer Society; 2015. p. 234–241.
[14] Fulda J, et al. TimeLineCurator: Interactive Authoring of Visual Timelines from Unstructured Text. IEEE Trans Vis Comput Graph. 2016;22(1):300–309.
[15] Styler IV W, et al. Temporal Annotation in the Clinical Domain. Transactions of ACL. 2014;2:143–154.
[16] Jung H, et al. Building timelines from narrative clinical records: initial results based-on deep natural language understanding. In: Proceedings of BioNLP 2011 workshop. ACL; 2011. p. 146–154.
[17] Tannier X, Vernier F. Creation, Visualization and Edition of Timelines for Journalistic Use. In: Proceedings of Natural Language meets Journalism Workshop at IJCAI; 2016. .
[18] Navas-Loro M, Filtz E, Rodríguez-Doncel V, Polleres A, Kirrane S. TempCourt: evaluation of temporal taggers on a new corpus of court decisions. The Knowledge Engineering Review. 2019;34:e24.
[19] Medvedeva M, Vols M, Wieling M. Using machine learning to predict decisions of the European Court of Human Rights. Artificial Intelligence and Law. 2020;28(2):237–266.
[20] Registry of the Court. European Court of Human Rights; 2020. Accessed 2020-09-14. `https://www.echr.coe.int/documents/rules_court_eng.pdf`.
[21] Chalkidis I, Fergadiotis M, Malakasiotis P, Androutsopoulos I. Large-Scale Multi-Label Text Classification on EU Legislation. CoRR. 2019;abs/1906.02192.
[22] Shaheen Z, Wohlgenannt G, Filtz E. Large-scale legal text classification using transformer models. In: Semapro 2020; to appear.. .
[23] Tuggener D, von Däniken P, Peetz T, Cieliebak M. LEDGAR: a large-scale multi-label corpus for text classification of legal provisions in contracts. In: 12th Language Resources and Evaluation Conference (LREC) 2020. European Language Resources Association; 2020. p. 1228–1234.
[24] Akbik A, Blythe D, Vollgraf R. Contextual String Embeddings for Sequence Labeling. In: COLING 2018, 27th International Conference on Computational Linguistics; 2018. p. 1638–1649.
[25] Filtz E, Kirrane S, Polleres A, Wohlgenannt G. Exploiting EuroVoc's Hierarchical Structure for Classifying Legal Documents. In: OTM Confederated International Conferences" On the Move to Meaningful Internet Systems". Springer; 2019. p. 164–181.
[26] Quemy A. European Court of Human Right Open Data project. CoRR. 2018;abs/1810.03115.
[27] Devlin J, Chang MW, Lee K, Toutanova K. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In: NAACL-HLT (1); 2019. p. 4171–4186.
[28] Sanh V, Debut L, Chaumond J, Wolf T. DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter. CoRR. 2019;abs/1910.01108.
[29] Lafferty JD, McCallum A, Pereira FCN. Conditional Random Fields: Probabilistic Models for Segmenting and Labeling Sequence Data. In: Brodley CE, Danyluk AP, editors. Proceedings of the Eighteenth International Conference on Machine Learning (ICML 2001). Morgan Kaufmann; 2001. p. 282–289.
[30] Leitner E, Rehm G, Moreno-Schneider J. Fine-grained Named Entity Recognition in Legal Documents. In: International Conference on Semantic Systems. Springer; 2019. p. 272–287.
[31] Chen D, Manning CD. A fast and accurate dependency parser using neural networks. In: Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP); 2014. p. 740–750.
[32] Navas-Loro M, Rodríguez-Doncel V. Annotador: a temporal tagger for Spanish. Journal of Intelligent & Fuzzy Systems. 2020;39:1979–1991. 2.
[33] Wolf T, Debut L, Sanh V, Chaumond J, Delangue C, Moi A, et al. HuggingFace's Transformers: State-of-the-art Natural Language Processing. arXiv e-prints. 2019 Oct;p. arXiv:1910.03771.
[34] Vaswani A, Shazeer N, Parmar N, Uszkoreit J, Jones L, Gomez AN, et al. Attention is all you need. In: Advances in neural information processing systems; 2017. p. 5998–6008.
[35] Cunningham H, et al. Getting More Out of Biomedical Documents with GATE's Full Lifecycle Open Source Text Analytics. PLOS Computational Biology. 2013 02;9(2):1–16.